

Laboratory 2: Postlab

Date 15/09/2022 Section _____
Name Anudeep Gadige

Addressing in MIPS (continued)

In this exercise you will continue exploring addressing in MIPS. You are also required to detail the memory model SPIM implements.

Step 1

As you could see during the inlab exercise, one can not run on the bare machine code that uses the extended instruction set.

Based on *lab2.2.asm* create a new program that will do the same thing but will be able to run on the bare machine. Save this as *lab2.3.asm*. Optimize your code as much as possible.

Q 1:

What is the number of instructions executed? Count only instructions between the label 'main' and the last instruction executed from your program.

Instruction Count = 15

Step 2

Create the program *lab2.4.asm* as follows:

- reserve space in memory for two variables called *var1* and *var2* of size word. The initial values of these variables will be the first two digits of your SSN for *var1* and the next two digits of your SSN for *var2*
- also reserve space in memory for two variables called *ext1* and *ext2* of size word. Use the '.extern' declaration for these two variables. The assembler will reserve space for them in the data segment that can be accessed using the *\$gp* register
- the program copies the values of *var1* and *var2* in *ext2* and *ext1* respectively
- use registers *\$t0* to *\$t8* if you need to
- use the extended instruction set
- each line in the program has a comment indicating what the instruction does

Q 2:

What are the displacements of *ext1* and *ext2* from the global pointer (**\$gp**) value?

Variable	Displacement (decimal)	Displacement (hexadecimal)
<i>ext1</i>	32768	0x00008000
<i>ext2</i>	32764	0x00007FFC

Q 3:

What exactly are the addresses where variables are stored in memory?

Variable	Address (hexadecimal)
<i>var1</i>	0x0000000c
<i>var2</i>	0x0000000a
<i>ext1</i>	0x10000000
<i>ext2</i>	0x10000004

Q 4:

How many native instructions are needed for each of the following memory accesses?

Memory Access	Native Instructions
<code>lw \$t0, var1</code>	<code>lui \$1, 4097</code>
	<code>lw \$8, 0(\$1)</code>
<code>sw \$t0, ext1</code>	<code>sw \$8, -32768(\$28)</code>

Step 3

Return to your lab instructor copies of *lab2.3.asm* and *lab2.4.asm* together with this postlab description. Ask your lab instructor whether copies of programs must be on paper (hardcopy), e-mail or both.